جافا سكريبت
دليل المبتدئين
JavaScript
A Beginner's Guide

مجلاد مشاري السبيعي

E-Mail
magedxl@hotmail.com

مقدمة إلى JavaScript

تعريف لغة JavaScript ؟
هي لغة كتابة نصوص برمجية لجانب المستضيف وتعتمد الكائنات ويمكنك استخدامها لجعل صفحات الويب أكثر ديناميكية.

أتت JavaScript كجهد مشترك بين شركة Netscape وشركة Sun , لقد تـم إعـلان اللغـة الجديـدة فـي العـام 1995 , إن لغـة كتابـة نصـوص برمجيـة لجانب المستضيف تعتمد الكائنات ويمكنك استخدامها لجعل صفحات الوب أكثر ديناميكية .

ملاحظه/
يـتم تشـغيل لغـة JavaScript مباشـرة عبـر المستضـاف الـذي يسـتخدمه المتصـفح وهـو برنـامج اسـتعراض الـوب دون الحاجـة لوسـيط أو خطـوات إضافية لإرسال واستخراج المعلومات من ملقم الوب.

* الإطار أو الشكل العام لـ JavaScript

```
<html>
<body>
<script language="JavaScript">
……..
</script>
</body>
```

يخبر هذا النص بان ما يلى هو JavaScript

نهاية JavaScript

وضع JavaScript في ملف HTML

* استخدام الوسوم Script في HTML
يتم استخدام الوسوم Script لأخبـار برنـامج الاسـتعراض أين ستبدأ بعض أنواع لغات كتابة النص البر مجي وأين ستنتهي في مستند HTML , في أكثر أشكالها شيوعا , تظهر الوسوم Script مثل اى مجموعة أخرى من الوسوم HTML .

```
<script>
JavaScript code here
</script>
```

البداية

النهاية

ملاحظه/
بعكس لغة HTML فأن لغة JavaScript حساسة لحالة الأحرف .

- إخفاء شيفرة JavaScript عن برامج الاستعراض الأقدم

ما الذي يحصل عندما يزور احدهم صفحتك باستخدام برنامج استعراض قديم لا يعتمد JavaScript ؟ .. إذا لم تكن حريصا قد يتم إظهار المحتويات الكلية لشيفرة JavaScript لديك على الصفحة كنص واضح , لمنع حصول هذا الأمر يمكنك إخبار برنامج الاستعراض لكي يتجاهل ما يلي بوضع تعليقات على HTML بين وسمي Script للفتح والإغلاق.

```
<script language="JavaScript">
<!--
JavaScript code here
//-->
</script>
```

بداية تعليق الإخفاء الشيفرة

نهاية تعليق الإخفاء الشيفرة

[illegible] HTML [illegible]
التعليق في JavaScript .

- إنشاء نصك البر مجي الأول

الآن وقـد أصـبحت تعـرف كيـف تسـتخدم الوسـوم Script فـي HTML لأخبـار برنـامج الاسـتعراض عـن JavaScript فـي مسـتند , سـنقوم حاليـا بكتابة النص Hello World.

```
<script language="JavaScript">
<!--
document.write("Hello World");
//-->
</script>
```

مـا نـراه أعـلاه هـو كتابـه للـنص Hello World باسـتخدام الدالـه document.write والعلامة ; في نهاية السطر البر مجي .

- إعادة زيارة Hello World في نموذج خارجي

افترض الآن بأنك ترغب باستخدام النص البر مجي Hello World على أكثر من صفحة واحدة لكنك لا ترغب بكتابته على كل صفحة , يمكنك فعل ذلك بوضع النص البر مجي في ملف نص برمجي خارجي واستدعائه بواسطة السمة src للوسم Script

قم بإنشاء الملف التالي:

```
document.write("Hello World");
```

وادرج الشيفرة أعلاه كما هو دون الوسوم Script أو تعليقات HTML
ثم قم بحفظها باسم jsfile1.js في محرر النصوص لديك .
بمجرد إنشاء حفظ الملف قم بإنشاء ملف HTML كما هو موضح أدناه:

```
<html>
<body>
<script language="JavaScript" src="jsfile1.js">
</script>
</body>
</html>
```

أمر الاستدعاء

- استخدام التعليقات في JavaScript

قد تحتاج لوضع ملاحظات في شيفرة JavaScript لديك , مثل وصف ما يفترض أن يقوم به النص البر مجي .
ولعمل ذلك كالتالي:

```
// Your Comment Here
```

في التنسيق أعلاه , سيتم تجاهل أي شي يسبق الشرطتين أعلاه . وهي تتبع نفس طريقة التعليق في لغة Visual Basic.

- إضافة تعليقات متعددة الأسطر

وهي موضحه كالتالي:

```
/*
My Script Will Write Some Code
Into HTML Document
*/
document.write("You Can See Me");
```

.JavaScript

ملاحظه/

يمكن أن تساعدك التعليقات في إزالة العلل من شيفرة JavaScript لديك .

استخدام المتغيرات

الآن وقد تعلمت أساسيات إضافة JavaScript إلى صفحة الوب , حان الوقت للتعرف أكثر على الأمور الداخلية للغة .

ما هي المتغيرة ؟

تمثل المتغيرة أو تحمل قيمة , يمكن أن يتم تغيير قيمة متغيرة في أي وقت , لفهم ما هية المتغيرة , خذ بعين الاعتبار العبارة الأساسية التالية التي قد تذكرها في لغة أخرى كـ C

```
X=2
```

يتم استخدام الحر فx كاسم للمتغيرة , ويتم اسنادة القيمة 2 إليه , لتغير القيمة أعطه:

```
X=4
```

- التصريح عن المتغيرات

للتصريح عن المتغيرات استخدم الكلمة الأساسية var التي تخبر البرنامج أن النص الذي يليه هو متغير .

```
Var coolcar;
```

حاليا لا تمتلك المتغيرة قيمة ولإسناد قيمة لها استخدم العبارة التالية:

```
Var coolcar=1200;
```

قم باستخدام عامل الإسناد = أعلاه لإسناد قيمة 1200 للمتغير coolcar .

- استخدام حالة الأحرف في المتغيرات

أن المتغيرات في لغة JavaScript حساسة لحالة الأحرف MY و my و My و mY هي متغيرات مختلفة بالنسبة للحروف الصغيرة والكبيرة . فان لغة JavaScript تعتبرها متغيرات جديدة ومختلفة عن بعضها البعض .

- استخدام الأحرف المسموح بها

هناك قاعدة مهمة يجب تذكرها وهي أن اسم المتغيرة يجب أن يبدأ بحرف عادي أو حرف تسطير (_) , لا يمكن أن يبدأ اسم المتغيرة برقم أو اى رمز أخر .

ملاحظه/
الفراغات البيضاء ممنوعة في أسماء المتغيرات.

متغيرات صحيحة :
Paycheck
_ Paycheck
Pay2check
Paycheck
Pay_255

متغيرات خاطئه للأسباب المبينة أدناه:
#Paycheck ← لأنه بدا برمز #
1Paycheck ← لأنه بدأ برقم
Pay check ← لوجود فراغ
_Pay check ← لوجود فراغ

- تجنب الكلمات المحجوزة

هنـاك قاعـدة أخـرى تـذكرها عنـد تسـمية متغيراتـك وهـي تجنـب اسـتخدام الكلمات المحجوزة فـي لغـة JavaScript , هنـاك كلمـات وأسماء محجـوزة مستخدمه من اجل غرض محدد في لغة JavaScript .

Abstact	Delete	Function	Null	Throw
Boolean	do	Goto	Package	Throws
break	Double	if	Private	Transient
Byte	Else	Implements	Protected	true
Case	Enum	Import	Public	Try
Catch	Export	in	Return	Typeof
New	For	Int	Short	Var
Const	Long	False	Char	While
continue	Switch	Final	Default	float

جدول الكلمات المحجوز

وهناك الكثير من الكلمات المحجوزة لم يسعفني ذكرها .

ملاحظه/
جميع الكلمات المحجوزة تكتب بحروف صغيرة .

- أنواع المتغيرات

1- الأرقام

ولتعريف المتغيرة الرقمية استخدم الكلمة الأساسية var :

```
Var max=number;
```

فالكلمة number تدل على أن المتغير max سيحجز قيمة رقمية .

```
Var max=1200;
Var max=29.99;
Var max=-24.5;
```

فالكلمة number لا تفرق بين الإعداد الصحيحة والحقيقة فكلها سواء (أرقام) .

2- النصوص

أن متغيرات السلاسل هي متغيرا سلسلة نصية قد تحتوى النصوص على فراغات أرقام حروف وما شابه فهي بين علامات تنصيص (" ")

```
Var mycar="stringtext";
```

```
Var mycar="corvette";
Var oldcar=" it is Toyota";
Var mycomputer="Pentium III, 700 MHz, 64MB Ram";
```

كما نرى يمكننا أن نكتب الأحرف صغيرة كبيرة أو أرقام وفراغات ورموز داخل السلاسل .

ملاحظه/

لا فرق بين علامتي التنصيص (" ") و (' ') فكلاهما سوا في الأمثلة أعلاه .

.

3- القيم المنطقية Boolean

أن المتغيرة المنطقية هي متغيرة مع قيمة تكون صح أو خطأ , فيما يلي بعض الأمثلة:

```
Var codes=true;
Var cool=false;
```

لاحظ بأنه لا حاجة لاحاطة الكلمتين بعلامة تنصيص , لانهما كلمتان محجوزتان في لغة JavaScript كقيم منطقية .
أيضا عوضا عن استخدام الكلمتين تسمح لك لغة JavaScript باستخدام 1 بدلا من true و 0 بدلا من false

```
Var codes=1;
Var cool=0;
```

4- القيمة Null

تعني Null بان المتغيرة لا تمتلك اى قيمة , إنها ليست فارغة ولا صفر , إنها ببساطة لا شي , إذا كنت تحتاج لتعريف المتغيرة مع قيمة Null , استخدم التصريح التالي:

```
Var codes=null
```

أن المتغيرة Null مفيدة عندما تختبر الدخل في النصوص البرمجة .

- استخدام المتغيرات في النصوص البرمجة

أنشي الصفحة التالية:

```
<html>
<body>
<script language="JavaScript">
<!--
var chipscots=2.59;
var istrue=false;
var nada=null;
document.write("maglad said, \"this project is fun\"");
//-->
</script>
</body>
</html>
```

- استدعاء المتغيرات

```
<html>
<body>
<script language="JavaScript">
<!--
var mycar="corvette";
document.write(mycar);
//-->
</script>
</body>
</html>
```

فقط محاط بقوسين أن نتيجة هذا البرنامج ببساطة كتابة الاسم corvette إلى برنامج الاستعراض .

- إضافة المتغيرات إلى سلاسات نصية

إذا أردت أن تتم طباعة تلك المتغيرة في المثال السابق مع بعض النص الأخر في سلسلة فاتبع الأتي:

```
<script language="JavaScript">
<!--
var mycar="corvette";
document.write("I like driving my "+mycar);
//-->
</script>
```

لجعل الشيفرة أسهل للكتابة , وضع كل سلسلة متضمنة في متغيرة , بحيث تحتاج فقط لإضافة قيم المتغيرات مع بعضها البعض بدلا من التعامل مع علامات الاقتباس كما يلي:

```
<script language="JavaScript">
<!--
var firststring="I like driving my ";
var mycar="corvette";
var secondstring=" every day !";
document.write(firststring+mycar+ secondstring);
//-->
```

تقوم هذه الشيفرة أعلاه بطباعة الجملة نفسها ولكنها تسمح لك بتغير أجزائها لاحقا بدون الحاجة لتحرير أمر القراءة.

شاهد المثال التالي للتعرف أكثر على المزيد من التطبيقات :

تم التصريح عن كل المتغيرات وإسناد قيم إليها

تمت طباعة قيم المتغيرات إلى برنامج الاستعراض هنا

```
<html>
<body>
<script language="JavaScript">
<!--
var hedingtext="<h1>a page of JavaScript<h1>";
var myintro="hello, welcome to my JavaScript page !";
var linktag="<a href='http://www.msn.com'>link to site</a>";
var redtext="<font color='red'>I am so good today !</font>";
var begineffect="<I>";
var endeffect="</I>";
var newssection="<P>";
document.write(hedingtext);
document.write(begineffect+ myintro+ endeffect);
document.write(newssection);
document.write(linktag);
```

استخدام الدالات (Functions)

ما هي الدالة ؟
بشكل أساسي , الدالة هي نص برمجي صغير ضمن نص برمجي اكبر , أن غرضها هو أداء مهمة منفردة أو سلسلة من الهام , يعتمد ما تقوم به الدالة على الشيفرة التي تضعها ضمنها , على سبيل المثال قد تكتب دالة سطر من النص إلى برنامج الاستعراض أو تحسب قيمة رقمية وتعيد تلك القيمة إلى النص البر مجي الأساسي.

- البناء والتصريح عن الدالات

```
Function functionname( )
```

أن الكلمة المحجوزة function تخبر برنامج الاستعراض بأنك تصرح عن دالة وان هناك معلومات إضافية ستليها , الجزء الثاني من المعلومات هو اسم الدالة . بعد ذلك , هناك مجموعة من الأقواس تدل فيما إذا كانت الدالة تقبل اى بارامترات.

على سبيل المثال سم الدالة realycool

```
Function realycool ( )
```

بما أن الدالة لا تستخدم اى بارمترات يتم ترك القوسين فارغين .

ملاحظه/
لا يتم استخدام نقطة منقوطة في نهاية الشيفرة.

- تعريف شيفرة الدالة

يحيط قوسان حاصران متعرجان ({}) بالشيفرة ضمن الدالة , يدل القوس المتعرج الأول على بداية شيفرة الدالة , بعد ذلك , تأتي الشيفرة . وفي النهاية يدل القوس المتعرج الثاني على نهاية الدالة , بالتنسيق التالي:

```
Function realycool ( )
{
JavaScript code here
}
```

سينفذ برنامج الاستعراض كل ين
المتعرجين عندما يتم استدعاء الدالة .
عندما يصل برنامج الاستعراض إلى القوس الحاصر المتعرج للإغلاق , فانه يعرف بأن الدالة قد انتهت .

- تسمية الدالات

كما هو الحال مع المتغيرات , يجب أن تتم تسمية الدالات بحرص لتجنب المشاكل في نصوصك البرمجة , أن القواعد الأساسية التي تنطبق على المتغيرات تنطبق نفسها على تسمية الدالات:
الحساسية لحالة الأحرف , استخدام الأحرف المسموح بها , تجنب الكلمات المحجوزة .

ملاحظه/
1- يجب أن يبدأ اسم الدالة بحرف عادي أو حرف تسطير (_)
2- لا يمكن أن يحتوى اسم الدالة على اى فراغات.

على سبيل المثال , افترض بأنك تنشى دالة تكتب بعض النص على صفحة , يمكن أن تحتوى على السطر التالي من الشيفرة :

```
document.write("<B>this is a bold statement !</B>");
```

يمكنك أن تسمى الدالة Text فقط , ولكن قد لا يكون هذا الاسم موصفا بما فيه الكفاية لأنه يمكن أن يكون هناك دالات أخرى تكتب نص أيضا إلى الصفحة , عوضا عن ذلك قد تسميها شيئا مثل print_bold_text وبالتي تعرف بان الدالة مستخدمة لطباعة جزء من النص بالخط الأسود العريض إلى برنامج الاستعراض , تظهر الدالة كالتالي:

```
Function print_bold_text( )
{
document.write("<B>this is a bold statement !</B>");
}
```

- إضافة بارامترات إلى الدالات

يتم استخدام البارامترات للسماح لدالة باستيراد قيمة أو أكثر من مكان ما خارج الدالة يتم تعيين البارامترات على السطر الأول من الدالة ضمن مجموعة الأقواس بالتنسيق التالي:

```
Function Functionname ( variable1,varubale2 )
```

أن اى قيمة يتم استحضارها كبارامتر تصبح متغيرة ضمن الدالة وتحمل الاسم الذي تعطيه لها ضمن القوسين .
على سبيل المثال , نجد فيما يلي كيفية تعريف الدالة realycool مع البارمترين المتغيرتين coolplace.coolcar :

```
Function realycool (coolplace, coolcar)
{
JavaScript code here
}
```

لاحظ بأنك في JavaScript , لا تستخدم الكلمة الأساسية var عندا تعين البارمترات من اجل دالة .

- استخدام قيم بارامترات الدالة

عندما تسند بارامترات إلى دالة , يمكنك استخدامها مثل اى متغيرات أخرى , على سبيل المثال , يمكنك إعطاء قيمة المتغيرة coolcar إلى متغيرة أخرى باستخدام عامل الإسناد كما في المثال التالي:

```
Function realycool (coolplace,coolcar)
{
var mycar=coolcar;
}
```

يتم إسناد قيمة coolcar إلى المتغيرة mycar

هذه الشيفرة أعلاه تسند قيمة البارامتر coolcar إلى متغيرة تدعى mycar عوضا عن إسناد قيمته إلى متغيرة أخرى , يمكنك استخدام البارامتر coolcar في الدالة , كما في المثال :

```
Function realycool (coolplace,coolcar)
{
document.write("my car is a "+coolcar);
}
```

تم استخدام قيمة المتغيرة coolcar في أمر القراءة أعلاه , إذا كانت قيمة coolcar هي Corvette , فستكون الدالة في السطر التالي عندما يتم استدعاؤها :

```
My car is a Corvette
```

- استخدام عدة بارمترات

فيما يلي تجد كيف يمكنك تغيير الدالة لاستخدام كلا البارامترين:

```
Function realycool (coolplace,coolcar)
{
document.write("my car is a "+coolcar+" and I drive it to "+coolplace);
}
```

الآن إذا كانت قيمة coolcar هي corvette وقيمة coolplace هيRiyadh city فان الدالة ستكون بعد استدعاؤها كالتالي:

```
My car is a Corvette and I drive it to Riyadh city
```

ملاحظه/
يمكنك استخدام اى عدد من البارامترات حسب حاجة الدالة .

- استدعاء الدالة

شاهد المثال التالي لتتعرف أكثر على الدوال :

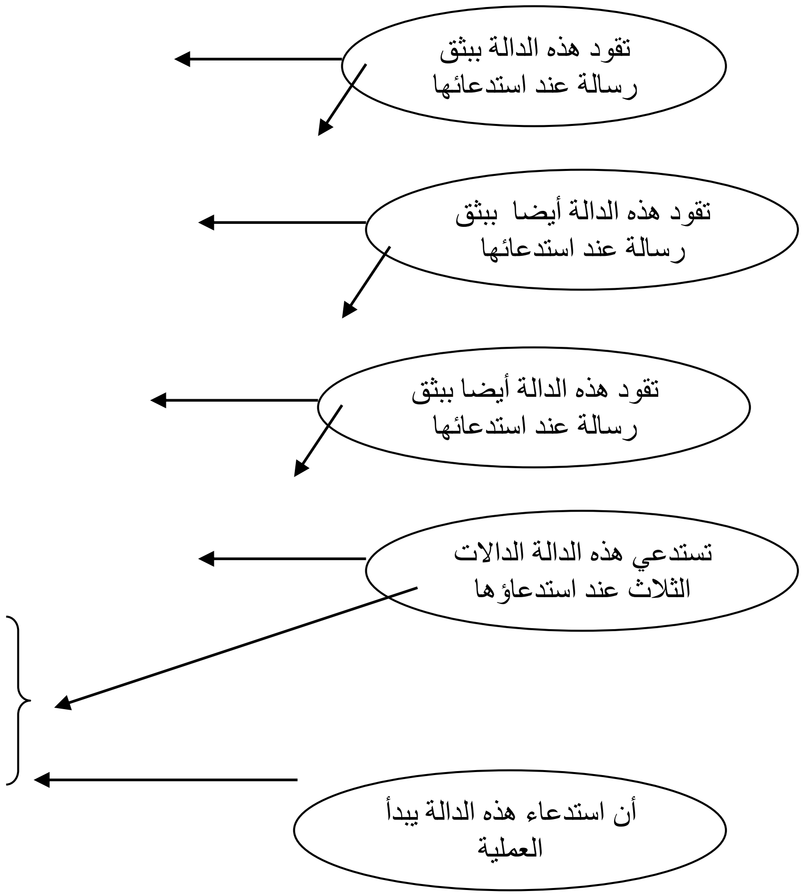
```
<html>
<head>
<title>function</title>
<script language="JavaScript">
<!--
function show_message( )
{
window.alert("this is an alert"
}
show_message( );
//-->
</script>
</head>
<body>
html code here
</body>
</html>
```

تسمية وبداية الدالة

يتم استدعاء الدالة وتنفيذها مما يؤدي إلى بثق مربع رسالة

أيضا شاهد المثال التالي كتوضيح معقد :

```
<html>
<head>
<title>more functions</title>
<script language="JavaScript">
<!--
function update_alert( )
```



ملاحظه/
يمكنك جعل ترتيب ظهور الدالات حسب برغبتك بإنشاء دالة وضمها لها كما في المثال أعلاه .

عوامل JavaScript

- أنواع العوامل:

1- العوامل الرياضية
2- عوامل الإسناد
3- عوامل المقارنة
4- العوامل المنطقية

1- العوامل الرياضية

بالنسبة للحاسب الرياضي , تستخدم عامل رياضي يمكن أن تكون القيم المستخدمة اى نوع تريده من القيم , كمثال سريع ستتذكر بأنك استخدمت عامل الجمع (+) لإضافة سلسلتين إلى بعضهما البعض في الوحدات السابقة كما يلي:

```
Window.alert(" I begin and "+"this is the end");
```

يمكن أيضا استخدام عامل الجمع عندما تكون إحدى القيمتين هي متغيرة كما في المثال التالي:

```
Var part2="this is the end"
Window.alert(" I begin and  "+part2);
```

- عامل الجمع

كما رأيت سابقا يمكن استخدام عامل الجمع من اجل تركيب سلسلتين , أيضا يمكنك استخدامها لجمع الأعداد في الحسابات الرياضية.

```
Var thesum=4+7;
Window.alert(thesum);
```

والنتيجة سيظهر لك رسالة بالمجموع وهو العدد 11 .

يمكنك جعل العملية أكثر تعقيدا بتحول احد العددين إلى متغيرة :

```
Var num1=4;
Var thesum=num1+7;
Window.alert(thesum);
```

تم إسناد العدد إلى متغيرة

لنطور المثال السابق قليلا يمكنك تحويل كلا العددين إلى متغيرتين:

```
Var num1=4;
Var num2=7;
Var thesum=num1+num2;
Window.alert(thesum);
```

تم جمع متغيرتين
باستخدام عامل الجمع

- عامل الطرح

يتم استخدام عامل الطرح لطرح القيمة الموجودة على الجانب الأيمن من القيمة الموجودة على الجانب الأيسر , كما هو الحال في الرياضيات:

```
Var thesum=10-2;
Window.alert(thesum);
```

والناتج 8 .

- عامل الضرب

يقوم عامل الضرب بضرب القيمتين على جانبي العامل ببعضهما البعض , كالضرب الرياضي:

```
Var num1=4;
Var num2=5;
Var thetotal=num1*num2;
Window.alert(thetotal);
```

والناتج هو القيمة 20 .

- عامل القسمة

يتم استخدام عامل القسمة لتقسيم القيم كالتالي:

```
Var num1=10;
Var num2=2;
Var theresult=num1/num2;
Window.alert(theresult);
```

ليظهر لنا الناتج 5 .

- عامل باقي القسمة

يتم استخدام عامل القسمة لإظهار العدد الباقي الصحيح كالتالي:

```
Var num1=11;
Var num2=2;
Var theresult=num1%num2;
Window.alert(theresult);
```

وستظهر لنا النتيجة وهي القيمة 1 وهي باقي القسمة الصحيحة .

- عامل الزيادة

يمكن استخدام عامل الزيادة على جانبي القيمة التي يعمل عليها , يزيد القيمة التي يؤثر عليها بمقدار 1 .

```
Var num1=2;
Var the result=++num1;
```

ملاحظة/

عند وضع num1++ فانه يزيد المعامل بقيمة مقدار واحد ثم ينفذها .

أما num1++ فانه يبدأ بالقيمة 2 أولا ثم يزيدها بمقدار 1 .

- عامل النقصان

مهمته هو إنقاص قيمة بمقدار 1 من القيمة التي يعمل عليها .

```
Var num1=2;
Var the result=--num1;
```

2- عوامل الإسناد

* عامل الإسناد (=)

لقد كنت تستخدم عامل الإسناد المباشر من قبل ويقوم هذا العامل بإسناد القيمة الموجودة على الجانب الأيمن للعامل إلى المتغيرة الموجودة على الجانب الأيسر كما يلي :

```
Var population=4500;
```

هذه الشيفرة أعلاه تسند القيمة 4500 إلى المتغيرة population .

- عامل الإضافة والإسناد (=+)

يقوم العامل =+ بإضافة القيمة الموجودة على الجانب الأيمن للعامل إلى المتغيرة الموجودة على الجانب الأيسر ثم يسند القيمة الجديدة إلى المتغيرة , كما يلي :

```
Var mymoney=1000;
Mymoney=mymoney+1;
```

تم أولا إسناد القيمة 1000 إلى المتغيرة mymoney , ثم بعد ذلك تسند إليها القيمة المتغيرة myomney زائدا القيمة 1
لتصبح النتيجة 1001 .

عوضا عن كتابة اسم المتغيرة مرة أخرى في السطر الثاني بامكانك عمل التالي:

```
Var mymoney=1000;
Mymoney+=1;
```

لتصبح النتيجة 1001 .

يمكنك أيضا عملها بطرق معقدة أكثر كما يلي :

```
Var mymoney=1000;
Var bouns=300;
Mymoney+=bouns;
```

تم إسناد القيمة لتصبح النتيجة 1300 .

وهناك أيضا العوامل المشابه للأعلى :

عامل الطرح والإسناد (-=)

عامل الضرب والإسناد (*=)

عام القسمة والإسناد (/=)

عامل الباقي والإسناد (%=)

3- عوامل المقارنة و العوامل المنطقية

* العوامل العلاقية

معناها	الأداة
اكبر من	>
اصغر من	<
اكبر من أو يساوي	>=

<=	اصغر من أو يساوي
= =	يساوي
!=	لا يساوي

- العوامل المنطقية

الأداة	معناها
&&	(حرف العطف و) And
||	(حرف العطف أو) Or
!	unary أداة أحادية (للنفي) Not

شاهد المثال التالي:

```
<Script language="JavaScript">
<!--
var num1=0;
var num2=0;
if (num1= =num2){
window.alert("true");
}
else
{
wind
}
//-->
</script>
```

العبارات والحلقات الشرطية

لشرط ما أو للقيام بشئ آخر إذا لم يتحقق الشرط .

- العبارة الشرطية If – Else

يمكننا إنشاء كتلة كاملة من الشيفرة , لنفترض بأننا نرغب بإرسال رسالة تقول you have the right number of boats إذا كانت المتغيرة boats=3 , وإذا لم تكن كذلك فنريد أن تظهر لنا الرسالة you do not have the right number boats .

```
if (boats= =3)
{
window.alert("you have the right number of boats");
}
else
{
window.alert("you do not have the right number of
boats");
}
```

المقارنة المعقدة .

```
If ((num1>2)&&(num1<<11))
{
window.alert("cool number");
}
else
{
window.alert("not a cool number");
}
```

ن

1 واقل من 11 .

- استخدام العبارة Switch

تسمح لنا العبارة Switch بأخذ قيمة منفردة وتنفيذ سطر مختلف من الشيفرة تبعا لقيمة المتغيرة .

```
Var thename="alsubai";

Switch (thename)
{
case "saud";
window.alert("saud is an ok nam
break;
case "alsubai"
window.alert("alsubai is the coo
window.alert("hi there alsubai");
break;
default;
window.alert("interesting name you have there");
}
```

تبدأ العبارة switch بالاعتماد على قيمة المتغيرة thename

هذه الحالة محققة وسيتم تنفيذها

يتم استخدام default عندما لا يكون اى من الحالات صحيحة

ملاحظه/

تخبر الجملة Break برنامج الاستعراض للخروج من كتلة الشيفرة والانتقال للسطر التالي من الشيفرة بعد الكتلة , ونستخدم العبارة Break في الكتلة Switch للتأكد من تنفيذ قسم case واحد فقط .

- الحلقة for

لنفرض أننا نريد تكرار العبارة I am part of loop 10 مرات بدون كتابتها نصيا .

```
For (count=1;count<11;count+=1)
{
document.write("I am part of loop");
}
```

سيتم تكرار هذا السطر 10 مرات

ملاحظه/

قم بكتابة الوسم
 بعد أمر قراءة النص لضمان أن تكون كل جملة في سطر جديد.

شاهد المثال التالي لإضافة أرقام للجمل:

```
For (count=1;count<11;count+=1)
{
document.write(count+".I am part of loop");
}
```

1.I am part of loop !
2.I am part of loop !
3.I am part of loop !
4.I am part of loop !
5.I am part of loop !
6.I am part of loop !
7.I am part of loop !
8.I am part of loop !
9.I am part of loop !
10.I am part of loop !

- الحلقة While

تنظر الحلقة While إلى مقارنة قصيرة وتكرر حتى تصبح المقارنة غير صحيحة للبدء انظر للمثال التالي :

```
<html>
<body>
<title>looping</title>
<body>
<script language="JavaScript">
<!--
var count=1;
while (count<11)
```

- الحلقة Do While

أن الحلقة Do While خاصة لأنه يتم تنفيذ الشيفرة ضمن الحلقة مرة واحدة على الأقل.

```
Var count=1;
Do
{
document.write("Hi!");
count+=1;
}
while (count<6);
```

معاملات الأحداث

- ما هو معامل الحدث ؟

إن معامل الحدث هو كلمة أساسية معرفة سابقا في JavaScript يتم استخدامه لمعاملة حدث على صفحة وب , قد تكون نقرة فأره أو نقر زر على الصفحة أو تحريك الفأرة فوق ارتباط على الصفحة .

- استخدام معاملات الأحداث

1- onClick

سنقوم حاليا باستخدام زر النقر onClick , واحد الأماكن الصالحة لكي يتم نقرها هو زر على النموذج , إذ دعنا نقول بأننا نرغب بإرسال شي ما إلى المستخدم عندما ينقر زر نموذج :

```
<html>
<body>
<form>
<input type="button" o[illegible]indow.alert('HI !');">
</form>
</body>
</html>
```

وسم فتح نموذج

وسم إغلاق نموذج

إذا أصبحت الشيفرة التي تريدها طويلة حقا , قد ترغب بوضع الشيفرة في دالة عوضا عن ذلك , ويمكن استخدامها لاستدعاء دالة عرفتها سابقا في الصفحة .

على سبيل المثال يمكننا وضع دالتين في القسم head لمستند html واستدعاء دالة من معامل حدث في القسم body تظهر الشيفرة :

```
<html>
<head>
<title>Events & Functions</title>
<script language="JavaScript">
<!--
function hi_and_salam( )
{
window.alert(' HI ');
window.alert(' Salam ');
}
//-->
</script>
</head>
<body>
<form>
<input type="button" onclick=" hi_and_salam( );">
</body>
</html>
```

تظهر هذه الدالة رسالتين إلى الشاشة

لاحظ كيف يتم استدعاء الدالة باستخدام معامل الحدث

2- onMouseOver

يحصل حدث مرور الفأرة عندا يحرك المستخدم مؤشر الفارة فوق نص ارتباط أو صورة مرتبطة أو جزء مرتبط من خريطة صورة .

```
<A href="http://www.msn.com"
onMouseOver="wondow.alert(' I told you not to try
click me !');">
don't try clicking me !</A>
```

ملاحظه/
يمكن أن يزعج الطريقة في المثال أعلاه الزوار لذلك لا يحبذا تكرراه في الصفحات .

OnLoad -3
يحدث هذا الحدث عندما يتم الانتهاء من تحميل الصفحة , ويتم إضافته للقسم body .

```
<Body onload="window.alert(' I ,m done loading now
!');">
</body
>
```

الكائنات

- ما هو الكائن ؟

الكائن هو طريقة لنمذجة شي حقيقي , حتى لو كان كينونة مجردة , عندما تفكر بالكائن , تتخيل على الأغلب أشياء عامة , كالسيارة مثلا وعندما ترى السيارة , تلاحظ أنها تمتلك مزايا عديدة , يمكننا أن ندعوها " خصائص " .
قد يوجد في السيارة راديو مع مشغل أقراص مضغوط CD , مقاعد جلدية ومحرك V-8 .
تشكل جميع هذه الأشياء أجزاء السيارة أو من منظور الكائنات خصائص كائن السيارة .

- إنشاء الكائنات

كما مع المتغيرات والدالات , توجد قواعد محددة يجب أن نراعيها عند تسمية الكائنات

1- حساسية حالة الأحرف .

2- تجنب الكلمات المحجوزة .

1- دالة المشيد

```
function car( )
{
Properties go here
}
```

تعريف دالة المشيد

هنا يتم سرد خصائص الكائن الذي ننشئة

نريد إنشاء كائن يدعى " car " مع الخصائص seats,theradio,engine .

```
function car(seats, engine, theradio )
{
thise.seats=seats;
this.engine=engine;
this.theradio=theradio;
}
```

دخل , وهي تطابق عدد الخصائص التي نريد أن يمتلكها الكائن , نلاحظ أيضا بأنه يتم إسناد قيم البارامترات إلى الخصائص التي نريد أن يمتلكها الكائن على كل حال يتم استخدام كلمة this اى بمعنى يمتلك هذا الكائن .

بعد إعداد خصائص الكائن باستخدام دالة المشيد , نحتاج لإنشاء ما يدعى " مثيل " للكائن لكي نتمن من استخدامه , ذلك لان دالة المشيد تنشى بنية الكائن فقط , ولا تنشئ مثيل للكائن قابل للاستخدام , من اجل إنشاء مثيل للكائن , نستخدم كلمة أساسية أخرى في لغة JavaScript وهي new .

```
var work_car=new car("cloth","V-6","Tape Deck");
```

الشـــي الأول الــذي ســنلاحظه هــو أننـــا ننشـــئ متغيــرة جديــدة تــدعى " work_car " , ستكون هذه المتغيرة مثيل جديد للكـائن car بسبب القيمـة التي نسندها إليها .
نلاحظ انه أيضا تم تمرير القيم إلى الدالة car كبارامترات , هذه القيم التـي نريد أن نستخدمها من اجل مثيل الكائن car المذكور , بمراعاة الترتيب

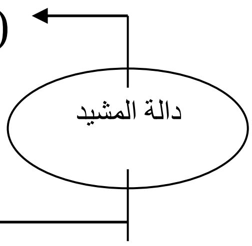
يمكننا أن نصل إلى المثيل work_car للكائن car . ويمكننا أن نعرف نوع المحـرك الـذي يملكـه work_car وذلـك بالوصـول إليـه باسـتخدام عامـل النقطة .

```
var engine_type=work_car.engine;
```

تسند هذه الشيفرة قيمة الخاصـية engine للمثيل work_car للكـائن car , إلــى المتغيـرة engine_type . وبمـا أننـا مررنـا V-6 كبـارامتر مـن اجـل المحــرك إلـــى دالـــة المشـــيد , يـــتم إســـناد القيمـــة V-6 إلـــى المتغيـــرة engine_type .

تجميـع الأجـزاء مـع بعضـها الـبعض : لكـي نتصـور العمليـة , دعنـا نجمـع الأجزاء الثلاثة مع بعضها لترى كيف تعمل لرؤية أسهل:

```
function car(seats, engine, theradio )
{
thise.seats=seats;
this.engine=engine;
this.theradio=theradio;
}
var work_car=new car("cloth","V-6","Tape Deck");
var engine_type=work_car.engine;
```



خصائص الكائن إلى متغيرة , عندما يتم إعداد المثيل work_car للكائن

car , يتم إسناد قيمة القماش إلى الخاصية work_car.seats , V-6 إلى الخاصية work_car.engine وهكذا .

لكي ترى كيف يعمل مثيل الكائن , دعنا نضيف مثيل أخر للكائن car إلى شفيرتنا , نستخدم الشيفرة التالي مثيلين للكائن car , أحداهما يدعى work_car والأخر يدعى fun_car :

```
Function car(seats, engine, theradio )
{
thise.seats=seats;
this.engine=engine;
this.theradio=theradio;
}
Var work_car=new car("cloth","V-6","Tape Deck");
Var fun_car=new car("leather","V-8","CD Player");
Var engine_type=work_car.engine;
Var seat_type=fun_car.seats;
```

الخاصية seats للمثيل fun_car للكائن car .

بعد القيام بذلك , نستطيع أن نكتب المزايا التي نريدها في سيارة مخصصة تجمع المزايا من كل أنواع السيارات , على سبيل المثال , انظر الشيفرة التالية , التي تكتب المزايا المطلوبة في سيارة مخصصة :

```
<html>
<head>
<script language="JavaScript">
<!--
function car(seats,engine,theradio )
{
this.seats = seats;
```

أن جزء النص
البرمجي مع المشيد أو
الدالة إنشاء المثيلات
واسنادات المتغيرات
الموجودة جميعا في
قسم head لكي يتم
تحميلها من البداية

عند طباعة النتيجة على برنامج الاستعراض ستكون كالتالي:

I want a car with leather seats.
I also need a V-6 engine.
OH and I would like a CD Player also.

2- ممهدات الكائن

أن ممهد الكائن اقصر بقليل من دالة المشيد , لكنه معتمد في JavaScript 1.2 والإصدارات الأحدث فقط . لذلك إذا قررت أن تستخدم هذا المنهج , فقد تحتاج لإضافة السمة "language="javascript1.2 إلى وسم الفتح <script> لكي لا يحصل الزوار الذين يستخدمون برامج استعراض قديمة على أخطاء .

دعنا ننشئ كان work_car باستخدام منهج الممهد , نريد أن يكون اسم الكائن work_car ولدية ثلاث مجموعات من الخصائص والقيم , تبين الشيفرة التالية كيف يتم ذلك :

```
work_car={seats:"cloth",engine:"V-6",theradio:"tape
deck"}
```

يمكننا أن نستخدم ممهد أخر من اجل الكائن fun_car أيضا , كما هو مبين في التالي:

```
work_car={seats:"cloth",engine:"V-6",theradio:"tape
deck"}
fun_car={seats:"leather",engine:"V-8",theradio:"CD
Player"}
```

يمكننا أن نكتب ما نريد من ميزات السيارة باستخدام هذه الخصائص , كما هو مبين في الشيفرة التالية :

```
<html>
<head>
<script language="JavaScript">
<!--
work_car={seats:"cloth",engine:"V-6",theradio:"tape
deck"}
fun_car={seats:"leather",engine:"V-8",theradio:"CD
Player"}
//-->
</script>
</head>
<body>
<script language="JavaScript">
<!--
document.write(" I want a car with "+ fun_car.seats+"
seats. <BR>");
document.write(" I also need a  "+work_car.engine+
"engine. <BR>");
document.write(" OH and I would lik
```

كائن الإطار Window

- مقدمة إلى كائن الأطر

يتم إنشاء كائن الأطر لكل إطار يظهر على الشاشة يمكن أن يكون الإطار إطار رئيسي ولقد استخدمنا في السابق العديد من الأطر كرسالة التنبية alert( ) والعديد .

1- المنهج confirm()

يمكن أن يتم استخدام المنهج لإعطاء المستخدم فرصة لتأكيد أو إلغاء عملية . يعيد هذا المنهج قيمة منطقية إما 0 أو 1 وبالتالي يتم إسناد نتيجته إلى متغيرة على الأغلب عند استخدامه.

فيما يلي التركيب النحوي له :

```
Var varname=window.confirm("your message");
```

شـاهد التطبيق التالي لتتعرف أكثر :

```
<html>
<head>
<script language="JavaScript">
<!--
function gothere( )
{
var is_sure=window.confirm("Are you want to leave
?");
if (is_sure= =true)
{
window.location="http://www.msn.com";
}
}
//-->
</script>
</head>
<form>
<input type="button" value="click to search the web"
onclick="gothere( );">
</form>
</body>
</html>
```

gothere( ) عند نقر الزر , ضمن الدالـة gothere( ) , نبـدأ بإسـناد القيمـة التـي يعيـدها مربـع التأكيـد إلـى متغيـرة اسـمها is_sure , إذا كانـت قيمـة is_sure مسـاوية لــ true يـذهب المسـتخدم للموقـع الجديـد أمـا إذا لـم تكـن كذلك فلا يحرك ساكنا .

2- المنهج print()

يسمح لك هذا المنهج للمستخدم بطباعة الإطار الحالي , عندما يتم استدعاء هذا المنهج , يجب أن يستحضر مربع حوار الطباعة للمستخدم لكي يعين المستخدم أعدادات الطباعة من اجل طباعة ا لمستند .

```
<form>
<input type="button" value="click to print the web"
onclick="window.print( );">
</form>
```

3- المنهج open()

يؤهلنا المنهج لفتح إطار جديد مع JavaScript , يأخذ المنهج ثلاث بارامترات ويعين البارامتر الثالث عدد من الخيارات التي قد يحتاجها الإطار.

```
Window.open("URL","name","attribulte1=value,attribut
e2=value");
```

شاهد التطبيق التالي لتتعرف أكثر :

```
<html>
<head>
<script language="JavaScript">
<!--
function matrix( )
{
window.open("a.html","matrix","width=400,height=300
,status=yes");
```

أيضا لوضع زر الإغلاق داخل النموذج الذي نريد استدعاءه نقوم بعمل التي داخل الصفحة المطلوب استدعاءها :

```
<input type="button" value="close window"
onclick="window.close( );">
```

مصفوفات JavaScript

- ما هي المصفوفة ؟

أن المصفوفة هي طريقة لتخزين بيانات من نفس الفئة لتأمين وصول سريع إليها من خلال النصوص البرمجية .

```
<html>
<head>
<script language="JavaScript">
<!--
var student0="ahmad ";
var student1="yousef ";
var student2="ali ";
var student3="Mohamed ";
var student4="Adam";
//-->
</script>
</head>
<body>
<script language="JavaScript">
<!--
document.write(student0+"<BR>");
document.write(student1+"<BR>");
document.write(student2+"<BR>");
document.write(student3+"<BR>");
//-->
</script>
</body>
</html>
```

- التسمية

يمكننا أن نسمي مصفوفة باستخدام القواعد نفسها التي تعلمناها في الوحدات السابقة .

- تعريف المصفوفة

يمكننا تعريفها كالتالي:

```
Var arrayname=new array(elemnt0,elemnt1);
```

```
Var s_list=new array(maged,ahmad,Thomas,salah);
```

أن رقم الفهرس هو الصفر لان المصفوفات تبدأ بالعد من 0 بدلا من 1 بالتالي , يجب الانتباه إلى هذا الأمر لتجنب وقوع مشاكل تتعلق برقم الفهرس للعنصر في المصفوفة , يمتلك العنصر الأول رقم الفهرس 0 والعنصر الثاني رقم الفهرس 1 والعنصر الثالث رقم الفهرس 2 وهكذا ...

- الوصول للمصفوفة

لنفترض الآن بأننا نريد أن نسند قيمة العنصر الأول في المصفوفة أعلاه إلى متغيرة تدعى tall_student , تذكر بان العنصر الأول يمتلك رقم الفهرس 0 :

```
<html>
<head>
<script language="JavaScript">
<!--
var s_list=new
Array("maged","ahmad","Thomas","salah");
var tall_student= s_list[0];
//-->
</script>
</head>
```

والنتيجة :

the tallest student in class is maged

- الطرق الأخرى لتعريف المصفوفات

إحدى الطرق لتعريف مصفوفة هي عن طريق إسناد مقدار معين من الفضاء (العناصر) إلى مصفوفة , ثم السماح بإسناد القيم لاحقا في النص البر مجي , يتم ذلك باستخدام رقم واحد ضمن القوسين عند تعريف المصفوفة :

```
var s_list=new Array(4);
```

تنشئ الشيفرة أعلاه مصفوفة تدعى s_list تمتلك أربع عناصر ويمكننا أن نضيف العناصر إلى المصفوفة لاحقا إذا رغبنا ذلك , تذكر أن أرقام الفهرس هي 3 , 2 , 1 , 0

ويتم الإسناد عن طريق الشيفرة التالية:

```
var s_list=new Array(4);
s_list[0]="maged";
s_list[1]="eike";
s_list[2]="naser";
s_list[3]="ahmad";
```

- المناهج

سننظر إلى المناهج التي يمكننا أن نستخدمها لإنجاز أشياء عديدة مع مصفوفاتنا .

1- المنهج ()concat

يتم استخدام المنهج أعلاه لجمع أو (ضم) عناصر مصفوفتين أو أكثر ويعيد مصفوفة جديدة تحتوى على جميع العناصر .

```
var fruits=new Array("oranges","apples");
var veggies=new Array("corn","peas");
var fruits_n_veggies= fruits.concat(veggies);
```

ملاحظه/

عند طلب الجمع نعمل مصفوفة جديدة باى اسم كالاسم أعلاه fruits_n_veggies .

2- المنهج join()

يتم استخدام المنهج أعلاه لجمع عناصر المصفوفة في سلسلة واحدة مع فصل كل عنصر بحرف أو رمز يتم ارسالة كبارمتر إلى المنهج إذا لم يتم استخدام منهج سيرسلها كفراغ حرفي.

```
var fruits=new Array("oranges","apples");
var fruits_string = fruits.join(:);
```

والنتيجة كالتالي:

Oranges : apples

3- المنهج ()reverse

يتم استخدام المنهج أعلاه لعكس ترتيب العناصر في المصفوفة , وبما أن عمله كذلك فلا داعي أن يتم إرسال بارامتر أو إعادة اى قيمة .

```
var fruits=new Array("oranges","apples");
fruits.reverse( );
```

والنتيجة كالتالي:

Apples oranges

4- المنهج ()sort

يرتب المنهج أعلاه المصفوفة حسب التسلسل الأبجدي وليس حسب الترتيب العددي , على سبيل المثال دعنا ننظر للمصفوفة التالية :

```
var fruits=new Array("oranges","apples","graoes");
fruits.sort( );
```

سوف تعيد هذه الشيفرة أعلاه ترتيب المصفوفة مرتبة حسب الترتيب الأبجدي :

Apples grapes oranges

- المصفوفات والحلقات

تسمح الحلقات بالتنقل عبر عناصر المصفوفة بدون الحاجة إلى التعامل مع كل عنصر على حدة بواسطة اسطر جديدة من الشيفرة , بدلا من ذلك , يمكننا أن نستخدم حلقة لكي نمرر عبر جميع عناصر المصفوفة ونختصر عدد الأسطر التي نحتاج لكتابتها من اجل المصفوفات الكبيرة .

- إنشاء عناصر المصفوفة

يمكن أن تكون الحلقة مفيدة في عملية إنشاء عناصر المصفوفة , وهي مفيدة بشكل خاص إذا كان المستخدم سيدخل محتويات المصفوفة لسبب ما أو إذا أردنا إنجاز حساب متشابه عند إنشاء كل عنصر.
لنفترض أننا نريد أن يكون المستخدم قادرا على إدخال أسماء الطلاب الأربعة من المصفوفة s_list لأسماء الطلاب باستخدام الحلقة for .

```
var s_list=new Array(4);
for(count=0;count<4;count++)
{
s_list[count]=window.prompt("enter a name"," ");
}
```

- التنقل عبر المصفوفة

يمكننـا أيضـا التنقـل عبـر المصـفوفة التــي تـم إنشــاؤها مـن اجـل تغييرهـا , الحصول على المعلومات منها أو سرد محتوياتها بالطريقة التي نريدها , أن ذلك مفيد جدا ويوفر علينا الوقت عند التعامل مع المصفوفات الكبيرة .

```
<html>
<head>
<script language="JavaScript">
<!--
var s_list=new
Array("maged","ahmad","Thomas","salah");
//-->
</script>
</head>
<body>
<H2>student names<H2/>
<script language="JavaScript">
<!--
for(count=0;count<4;count++)
{
document.write(s_list[count]+ "<BR>");
}
//-->
</script>
</body>
</html>
```

يتم إنشاء مصفوفة وانساد قيم إلى عناصرها

تتم طباعة المصفوفة باستخدام حلقات التكرار

ليتم إظهار النتيجة كالتالي:

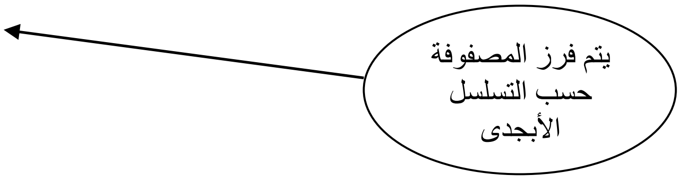
student names

maged
ahmad
Thomas
Salah

ملاحظه/
يمكننا أن نرى كيف أن الحلقات توفر علينا إدخال المعلومات عندما نقرر طباعة لائحة الطلاب على الشاشة .

لنفرض الان إننا نريد طباعة أسماء طلاب من خلال مصفوفة الحلقات لكن بالترتيب الأبجدي للأسماء :

```
<html>
<head>
<script language="JavaScript">
<!--
var s_list=new
Array("maged","ahmad","Thomas","salah");
```



الكائنان Math و Date

- الكائن Math

يمكن أن يكون الكائن Math مفيدا لنا عندما نحتاج لان نؤدي عمليات حسابية متنوعة في نصوصنا البرمجية .

- ما هو الكائن Math ؟

الكائن Math هو كائن معرف بشكل مسبق في لغة JavaScript , يقدم لنا خصائص ومناهج يتم استخدامها تماما مثل الكائنات الأخرى المعرفة بشكل مسبق .

ويتم استخدام الكائن Math من اجل الأغراض الرياضية لكي يقدم قيم بعض الثوابت الرياضية أو لتأدية عمليات محددة عندما نستخدم دالة منهج .

- الخصائص

الخاصية PI
تحمل هذه الخاصية القيمة الرقمية لـ PI , هي تقريبا ...3.14159
نستطيع أن نرى قيمة الخاصية PI باستخدام مربع رسالة , يعرض ما يلي مربع رسالة مع قيمة هذه الخاصة:

```
Window.alert(math.PI);
```

وسيعرض لنا رسالة ناتجة من الشيفرة أعلاه

3.14159

ملاحظه/
هناك العديد من الخاصيات لكن لن نذكرها للاختصار ..

- المناهج

المنهج sqrt()

سنستخدم المنهج كمثال . الطريقة الأسهل لاستخدام المنهج هي بوضع رقم موجب كبارامتر للمنهج , كما هو ظاهر في المثال التالي:

```
Window.alert(math.sqrt(4));
```

تعرض الشيفرة أعلاه قيمة الجذر التربيعي لــ4 على الشاشة وهي :

2

شاهد المثال التالي:

```
<html>
<head>
<script language="JavaScript">
<!--
function get_a_root()
```

تبدأ الدالة التي تقوم بالمهمة

يتم توجية المستخدم من اجل رقم , يتم اسنادة فيما بعد إلى متغيرة

عندما ندخل مثلا في المثال أعلاه العدد 16 فان الناتج سيكون:

4

- الكائن Date

الكائن Date هو كائن أخر معرف مسبقا في لغة JavaScript يسمح لنا بالتعامل مع الوقت والتاريخ للحصول على قيم وقت محددة نستطيع أن نستخدمها في نصوصنا البرمجية.

```
var rightnow= new Date( );
```

لقد أنشئنا مثيل للكائن Date أعلاه وهو rightnow ولك الحرية في إطلاق أسماء أخرى .

- المناهج

Getday( )
Getmonth( )
Getyear( )
Gettime( )
Getseconds( )
Getdate( )
Getminutes( )

.... الـــخ

- ما رأيك ببعض النصوص البرمجية للتاريخ ؟

اكتب التاريخ على الصفحة

لكـي نكتـب التـاريخ علـى الصـفحة , سنسـتخدم بعـض منـاهج الكـائن Date للحصـول علـى القيم التـي نريـدها , لنقـل أننـا نريـد كتابـة تـاريخ مـع التنسـيق المشـابة للشكل التالـي :

Tuesday. 11/21/2001

للقيام بـذلك نحتـاج لمعرفـة يـوم الأسـبوع , الشـهر , السـنة , ونسـتخدمها عن طريق المناهج التي ذكرناها سابقا .

Getday , getyear , ... الـــخ

```
<html>
<head>
<script language="JavaScript">
<!--
var rightnow= new Date();
var weekday= rightnow.getDay();
var themonth= rightnow.getMonth();
var thedate= rightnow.getDate();
var theyear= rightnow.getYear();
// see the days of the week
var someday= new Array(7)
someday[0]="Sunday";
someday[1]="Monday";
someday[2]="Tuesday";
someday[3]="Wednesday";
someday[4]="Thursday";
someday[5]="Friday";
someday[6]="Saturday";
// see the month number to be recognizable
themonth+=1;
// see the year date for 4 digits
if (theyear<2000)
theyear+=1900;
//-->
</script>
</head>
<body>
<H1>the Date</H1>
<script language="JavaScript">
<!--
document.write(someday[weekday]+ ",
"+themonth+"/"+thedate+"/"+theyear);
//-->
</script>
</body>
</html>
```

إنشاء مثيل للكائن

إسناد نتائج المناهج المستخدمة إلى متغيرات

إنشاء مصفوفة من اجل أيام الأسبوع

تعديل السنة على أربعة أرقام

النهاية

16 جمادى الأولى 1425